**HEAP SORT**

Sort a given set of N integer elements using Heap Sort technique and

compute its time taken.

#include <stdio.h>

#include <time.h>

clock\_t start, end;

void swap(int\* a, int\* b)

{

int temp = \*a;

\*a = \*b;

\*b = temp;

}

void heapify(int arr[], int N, int i)

{

int largest = i;

int left = 2 \* i + 1;

int right = 2 \* i + 2;

if (left < N && arr[left] > arr[largest])

largest = left;

if (right < N && arr[right] > arr[largest])

largest = right;

if (largest != i) {

swap(&arr[i], &arr[largest]);

heapify(arr, N, largest);

}

}

void heapSort(int arr[], int N)

{

for (int i = N / 2 - 1; i >= 0; i--)

heapify(arr, N, i);

for (int i = N - 1; i >= 0; i--) {

swap(&arr[0], &arr[i]);

heapify(arr, i, 0);

}

}

void printArray(int arr[], int N)

{

for (int i = 0; i < N; i++)

printf("%d ", arr[i]);

printf("\n");

}

int main()

{

int arr[1000];

int N;

printf("Enter number of elements:");

scanf("%d",&N);

printf("\nEnter elements:");

for (int i=0; i<N; i++)

{arr[i]=rand();}

start=clock();

heapSort(arr, N);

end=clock();

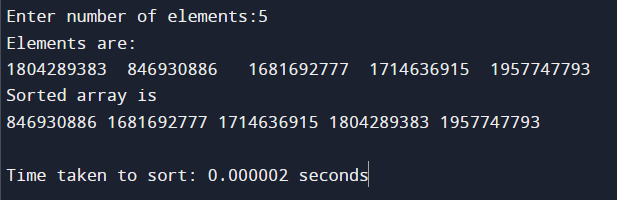
printf("Sorted array is\n");

printArray(arr, N);

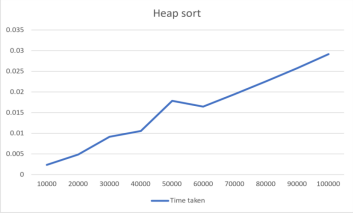
printf("\n Time taken to sort: %f seconds",(double)(end-start)/CLOCKS\_PER\_SEC);

}

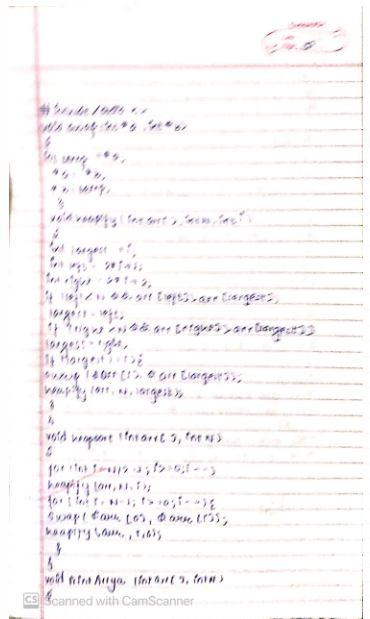
**OUTPUT:**



**GRAPH:**

****

**LAB RECORD:**

****

